This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (currently amended) A method for processing a query including an extensible markup language based expression with instructions to modify data that is stored in a node of an extensible markup language schema in a database, the method comprising:

transforming an abstract syntax tree corresponding to the expression into a unified tree including extensible markup language based algebra operations;

mapping the extensible markup language based algebra operations in the unified tree to enhanced relational algebra based extensible markup language modification operations in a relational tree; and

executing the query by modifying data that is stored in the node of the extensible markup language schema in the database in accordance with the relational tree comprising the enhanced relational algebra based extensible markup language modification operations.

2. (original) The method of claim 1, wherein transforming the abstract syntax tree comprises:

recursively traversing the abstract syntax tree;

generating a unified sub-tree for each abstract syntax tree node, the sub-tree including at least one corresponding extensible markup language based algebra operation; and

inserting the sub-tree into the unified tree.

3. (original) The method of claim 1, wherein mapping the extensible markup language based algebra operations comprises:

recursively traversing the unified tree;

generating a relational sub-tree for each unified tree node, the sub-tree including at least one corresponding relational algebra based operation; and

inserting the sub-tree into the relational tree.

4. (original) The method of claim 1, further comprising parsing the query to yield the extensible markup language based expression.

5. (original) The method of claim 1, further comprising parsing the extensible markup language based expression to yield the abstract syntax tree.

6. (original) The method of claim 1, further comprising generating a query plan according to the relational tree.

7. (original) The method of claim 6, further comprising submitting the query plan to a query processor for execution by the query processor.

8. (currently amended) The method of claim 1, comprising mapping the extensible markup language based algebra operations in the unified tree to enhanced relational algebra based extensible markup language modification operations with nested table abstraction in the relational tree.

9. (original) A computer readable medium having computer-executable instructions for performing the steps recited in claim 1.

10. (currently amended) A database engine for processing a query including an extensible markup language based expression with instructions to modify data that is stored in a node of an extensible markup language schema in a database, the database engine comprising:

an extensible markup language operation generator for transforming an abstract syntax tree corresponding to the expression into a unified tree including extensible markup language based algebra operations;

an extensible markup language algebrizer for mapping the extensible markup language based algebra operations in the unified tree to enhanced relational algebra based extensible markup language modification operations in a relational tree; and

a query processor that executes the query by modifying data that is stored in the node of the extensible markup language schema in the database in accordance with the relational

tree comprising the enhanced relational algebra based extensible markup language modification operations.

11. (original) The database engine of claim 10, further comprising a relational parser for parsing the query to yield the extensible markup language based expression.

12. (original) The database engine of claim 10, further comprising an extensible markup language parser for parsing the extensible markup language based expression to yield the abstract syntax tree.

13. (canceled)

14. (original) The database engine of claim 10, wherein the extensible markup language operation generator transforms the abstract syntax tree by recursively traversing the abstract syntax tree; generating a unified sub-tree for each abstract syntax tree node, the sub-tree including at least one corresponding extensible markup language based algebra operation; and inserting the sub-tree into the unified tree.

15. (original) The database engine of claim 10, wherein the extensible markup language algebrizer maps the extensible markup language based algebra operations by recursively traversing the unified tree; generating a relational sub-tree for each unified tree node, the sub-tree including at least one corresponding relational algebra based operation; and inserting the sub-tree into the relational tree.

16. (previously presented) A method for processing a query, the method comprising:

parsing the query to yield an extensible markup language based expression; and

generating a query plan for the expression including enhanced relational algebra expressions with a nested table abstraction operation; and

executing the query based on the query plan by using the nested table abstraction operation to establish a parent to descendent relationship among instances of nodes in an extensible markup language schema without compiling separate lists.

17. (original) The method of claim 16, comprising generating a query plan for the expression including enhanced relational algebra expressions with a nested table abstraction operation that is one of a row nesting operation, a nested table expansion, a nested row expansion, and a nested row descendant expansion.

18. (original) The method of claim 16, wherein generating the query plan comprises:

parsing the expression to yield an abstract syntax tree;

transforming the abstract syntax tree into a unified tree including extensible markup language based algebra operations; and

mapping the extensible markup language based algebra operations in the unified tree to relational algebra based operations in a relational tree.

19. (original) A computer readable medium having computer-executable instructions for performing the steps recited in claim 16.

20. (previously presented) A database engine comprising:

a relational parser for parsing a query to yield an extensible markup language based expression;

an extensible markup language algebrizer for generating a query plan for the expression including enhanced relational algebra expressions with a nested table abstraction operation; and

a query processor that executes the query based on the query plan by using the nested table abstraction operation to establish a parent to descendent relationship among instances of nodes in an extensible markup language schema without compiling separate lists corresponding to each of the nodes.

21. (original) The database engine of claim 20, wherein the nested table abstraction operation is one of a row nesting operation, a nested table expansion, a nested row expansion, and a nested row descendant expansion.

22. (original) The database engine of claim 20, further comprising:

an extensible markup language parser for parsing the expression to yield an abstract syntax tree; and

an extensible markup language operation generator for transforming the abstract syntax tree into a unified tree including extensible markup language based algebra operations.

23. (original) The database engine of claim 22, wherein the extensible markup language algebrizer maps the extensible markup language based algebra operations in the unified tree to relational algebra based operations in a relational tree.